Bauhaus-Universität
Weimar

Machine Learning Models for Topology Optimization

Submitted by,

Midhun Kanadan (124622)

Table of Contents

Table of Figures

1. Introduction

Topology optimization plays a crucial role in engineering design by enabling the creation of lightweight and structurally efficient solutions. Traditional optimization techniques have demonstrated their effectiveness, but their reliance on predefined design parameters and assumptions limits their ability to explore complex design spaces. In recent years, the integration of Machine Learning (ML) and Deep Learning (DL) techniques has emerged as a powerful approach to tackle topology optimization problems. By leveraging ML algorithms, engineers can now uncover innovative design solutions, optimize complex structures, and expedite the design process. This report provides an overview of ML models applied to topology optimization and highlights their potential to revolutionize engineering design.

1.1 Background and Problem Formulation

Optimization techniques have long been at the forefront of efforts to achieve efficient and sustainable structural design solutions. Among these techniques, topology optimization has emerged as a key approach for determining the optimal arrangement of structural components within a given design domain. Traditional methods, including Evolutionary Structural Optimization (ESO), Solid Isotropic Material with Penalization (SIMP), and level set techniques, have made significant contributions to the field. However, recent advancements in machine learning (ML) and deep learning (DL) present new opportunities to enhance the efficiency and effectiveness of topology optimization processes.

Topology optimization aims to generate structurally efficient designs that meet specific performance criteria while minimizing material usage and costs. Conventional approaches rely on finite element modeling and mathematical optimization algorithms, but they often face challenges when dealing with complex design problems, computationally expensive simulations, and uncertainties. This has led to the exploration of ML and DL as alternative approaches to address these challenges in topology optimization.

The development of topology optimization techniques has been the subject of extensive research over the years, resulting in notable contributions that have shaped the field. Works such as Ole Sigmund's 99-line topology optimization code and Liu and

Tovar's efficient 3D topology optimization code have made significant impacts. Sigmund's code provided a concise and practical implementation of topology optimization algorithms, enabling engineers and designers to optimize structures efficiently. Liu and Tovar's methodology, on the other hand, introduced a comprehensive approach to optimizing three-dimensional structures, expanding the capabilities of topology optimization.

These works, among many others, have laid the foundation for further advancements in topology optimization methodologies. They have demonstrated the effectiveness of various methods employed in topology optimization. The Evolutionary Structural Optimization (ESO) method pioneered the concept of removing inefficient material from designs, while the Bi-directional Evolutionary Structural Optimization (BESO) method extended this approach by allowing the material to be added as needed. The Solid Isotropic Material with Penalization (SIMP) method has been widely used to optimize material distribution by penalizing intermediate densities. Level set techniques have also emerged as a versatile framework for capturing and evolving geometric shapes during the optimization process.

Topology optimization has found applications in various industrial sectors, further emphasizing its importance and relevance. For instance, the leading-edge droop nose ribs for the Airbus A380 serve as a typical example of topology optimization applied in the aerospace industry. By utilizing topology optimization techniques, the weight of these critical components was significantly reduced, leading to improved fuel efficiency and overall aircraft performance.

Topology optimization is not limited to specific industries but finds widespread use in fields such as civil engineering, biochemical engineering, and mechanical engineering. By proactively incorporating topology optimization into their design processes, engineers in these fields can create innovative and efficient solutions that outperform traditional manual designs. This approach allows for improved structural performance, reduced material usage, and enhanced sustainability.

While traditional topology optimization methods have made significant strides in optimizing structures, recent advancements in machine learning (ML) and deep learning (DL) present new opportunities and challenges. ML and DL algorithms have shown tremendous potential in analyzing complex datasets, learning patterns, and making accurate predictions. In the context of topology optimization, ML and DL methods offer the prospect of revolutionizing the field by integrating historical data and advanced algorithms to model the intricate relationships between design parameters and structural responses.

Machine learning and deep learning models offer the potential to revolutionize topology optimization by leveraging extensive datasets, pattern recognition, and complex modeling capabilities. ML algorithms have the ability to learn from existing data and capture intricate relationships between input parameters and structural responses. This enables the development of predictive models for efficient optimization. DL, on the other hand, excels in learning hierarchical representations of data, enabling the discovery of complex design patterns and the generation of novel structural configurations. By incorporating ML and DL into topology optimization processes, enhanced design exploration, faster convergence, reduced computational costs, and improved reliability can be achieved.

Deep Convolutional neural networks (CNNs) have emerged as a powerful class of DL models, particularly well-suited for analyzing spatial features and patterns. Although CNNs may not have direct applicability to topology optimization, other ML and DL architectures, such as feedforward neural networks or recurrent neural networks, can be employed to capture the complex relationships within structural data. These models enable the development of predictive models that assist in generating optimized designs.

Furthermore, dimensionality reduction methods, such as Principal Component Analysis (PCA), prove invaluable in exploring the most influential design parameters. By reducing the dimensionality of the data, PCA facilitates a better understanding of the essential variables that significantly impact the structural response, thus reducing computational complexity and enabling more efficient optimization.

This research project aims to explore the potential of ML and DL models, in conjunction with dimensionality reduction techniques, in the field of topology optimization. Building upon the foundations laid by previous works, including methodologies such as ESO, BESO, SIMP, and level set techniques, this study seeks to develop and evaluate robust ML and DL models for topology optimization. By leveraging historical data, implementing advanced algorithms, and parameterizing the geometry, the project aims to improve the efficiency and effectiveness of topology optimization, leading to enhanced structural performance, cost reduction, and environmental sustainability.

In the subsequent sections of this report, we will delve into the details of topology optimization methodologies, machine learning and deep learning methods, previous works in the field, and the proposed methodology for implementing ML and DL models for topology optimization. The report will present the experimental setup, dataset description, model development, evaluation metrics, and a comprehensive discussion of the results. Additionally, we will explore potential future directions for research in this rapidly evolving domain.

1.2 Structure of the Research Report

Chapter 1: Introduction

The introduction chapter provides an overview of the research topic, the motivation behind using machine learning and deep learning models for topology optimization, and the objectives of the study. It outlines the significance of the research and highlights the potential benefits of integrating machine learning and deep learning techniques in topology optimization. The chapter also presents the structure of the report and briefly introduces the subsequent chapters.

Chapter 2: Literature Review

In this chapter, a comprehensive literature review is conducted to explore the existing body of knowledge on structural optimization, topology optimization, machine learning, and deep learning. It examines previous works and studies related to topology optimization methods, including traditional approaches such as the SIMP method, level set techniques, ESO, and BESO. The chapter also reviews relevant literature on the application of machine learning and deep learning models in topology optimization. It highlights the advancements, limitations, and contributions of these methodologies in the field.

Chapter 3: Methodology and Implementation

This chapter focuses on the methodology and implementation of machine learning and deep learning methods for topology optimization. It discusses the data preprocessing techniques employed, such as data normalization and feature extraction. The chapter elaborates on the selection and architecture of machine learning and deep learning models, including convolutional neural networks (CNNs) and Encoder Decoder models. It also describes the training process, including the choice of loss functions, optimization algorithms, and hyperparameter tuning.

Chapter 4: Results

In this chapter, the results of applying machine learning and deep learning models to topology optimization are presented and analyzed. It includes the evaluation of the models' performance in terms of optimization efficiency, accuracy, convergence, and robustness. The chapter showcases the obtained results through visual representations, statistical analysis, and comparisons with traditional optimization methods. It discusses the implications and significance of the results in advancing the field of topology optimization.

Chapter 5: Conclusion

The conclusion chapter summarizes the key findings of the study and draws conclusions based on the results and analysis. It reflects on the extent to which the research objectives have been achieved and provides insights into the contributions of the study. The chapter also discusses the limitations of the proposed methodology and suggests areas for future research and improvements in the integration of machine learning and deep learning models in topology optimization.

Chapter 6: Further Work

This chapter explores potential avenues for further research and development in the field of machine learning and deep learning models for topology optimization. It discusses possible extensions to the current study, identifies unresolved research questions, and proposes new directions for future investigations. The chapter emphasizes the importance of continuous exploration and refinement of machine learning and deep learning techniques to enhance the effectiveness and efficiency of topology optimization.

2. Literature Review

2.1 Structural Optimization

Structural optimization is a field of study that aims to find the best design configuration of a structure given certain objectives and constraints. It involves optimizing the layout, shape, and material distribution of a structure to achieve desired performance criteria, such as maximum stiffness, minimum weight, or improved structural integrity. Structural optimization plays a crucial role in engineering and design processes, as it enables the creation of efficient and cost-effective structures.

To formulate the structural optimization problem, several components need to be introduced. An objective function (f) is established to represent the objective to be minimized or maximized. Usually, in classical topology optimization problems, the objective function aims to minimize the compliance of the structure or its weight (volume, mass, cross-section). The design variables (x) define the structure's design, often associated with its geometry, while the state variables (y) represent the structural response, such as stress, strain, or displacement. The state variables are dependent on the design variables, forming a relationship between the design and the structural behavior.

A general structural optimization problem can be formulated as follows (Peter W. Christensen, 2009):

$$\text{Minimize } \mathrm{f}(\mathrm{x}, \mathrm{y}) \text{ with respect to x and y}$$

$$\text{Subject to} \begin{cases} \text{behavioral constraints on y} \\ \text{design constraints on x} \\ \text{equilibrium constraint.} \end{cases}$$

2.1.1 Types of Structural Optimization

In the field of structural optimization, various types of optimization techniques are employed to achieve material-efficient or cost-effective designs. These techniques can be broadly categorized into the following three types:

- Size Optimization

- Shape Optimization
- Topology Optimization

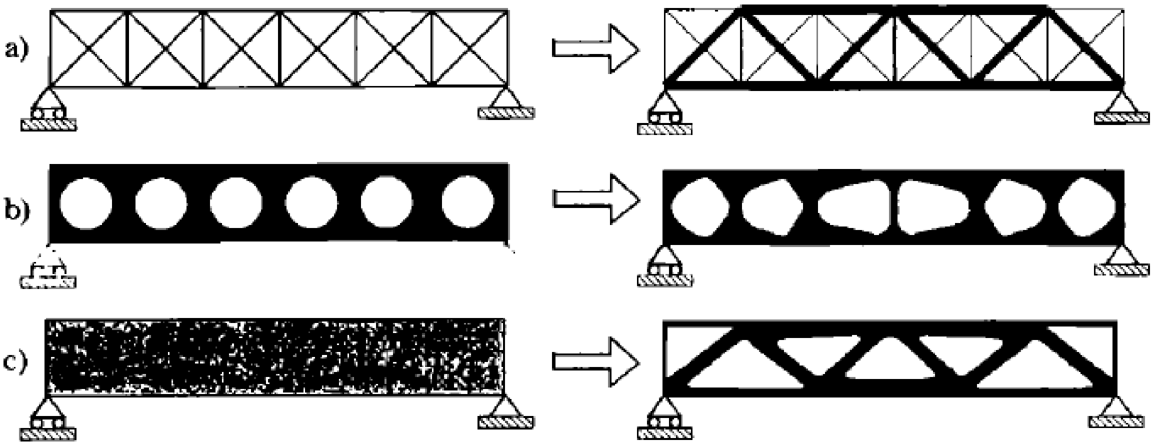


Figure 1: a) Sizing optimization of a truss structure, b)shape optimization and c) topology optimization (Bendsϕe, 2004)

2.1.1.1 Size Optimization

Size optimization focuses on determining the optimal dimensions of structural components, such as trusses, beams, and frames, as well as the thicknesses of membranes, plates, and shells. The objective of size optimization is to minimize the weight or material usage while satisfying specific design constraints.

In size optimization, the cross-sectional dimensions of trusses, beams, frames, or the thicknesses of membranes, plates, and shells are chosen as design variables. By varying these dimensions, designers can explore different design configurations and find the optimal combination that meets specified design constraints.

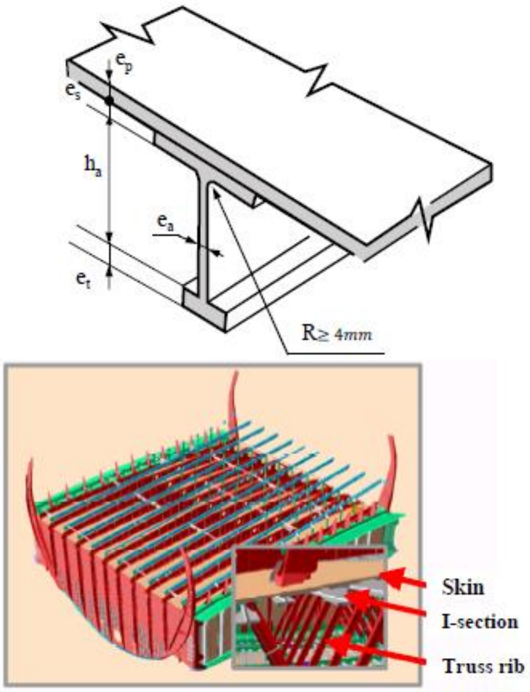


Figure 2: Sizing optimization for aircraft structure members(Weihong et al., 2016)

2.1.1.2 Shape Optimization

Shape optimization aims to achieve desired structural behavior, such as minimizing stress concentrations, improving load distribution, or enhancing structural stability, while satisfying specified constraints. It involves changing the contours, curves, or profiles of structural elements to optimize their response as shown in Figure 1.

In shape optimization, the shape of the structural components becomes the design variable. By altering the geometry, designers can explore various shapes and configurations to find the optimal one that meets the desired criteria. This optimization process often involves advanced mathematical algorithms and numerical techniques to iteratively refine the shape based on predefined objectives and constraints.

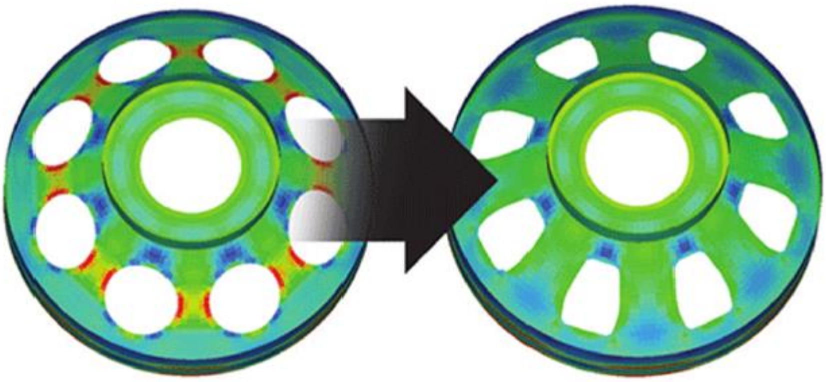


Figure 3: Shape optimization of cut-outs in a thin walled structure (Weihong et al., 2016)

2.1.1.3 Topology Optimization

Topology optimization is a specific approach within structural optimization that focuses on determining the optimal material distribution within a given design domain as shown in Figure 4: Topology optimization of MBB beam (Weihong et al., 2016)Figure 4. It seeks to find the most efficient configuration of materials that maximizes structural performance while satisfying specified constraints.

Unlike size optimization and shape optimization, which modify existing designs, topology optimization allows for the exploration of entirely new structural configurations. It considers the material layout itself as a design variable, enabling the discovery of novel and unconventional solutions.

The objective of topology optimization is to identify the arrangement of materials that achieves the desired structural objectives, such as maximizing stiffness, minimizing weight, or improving energy absorption. By optimizing the material distribution, topology optimization can lead to highly efficient and structurally robust designs.

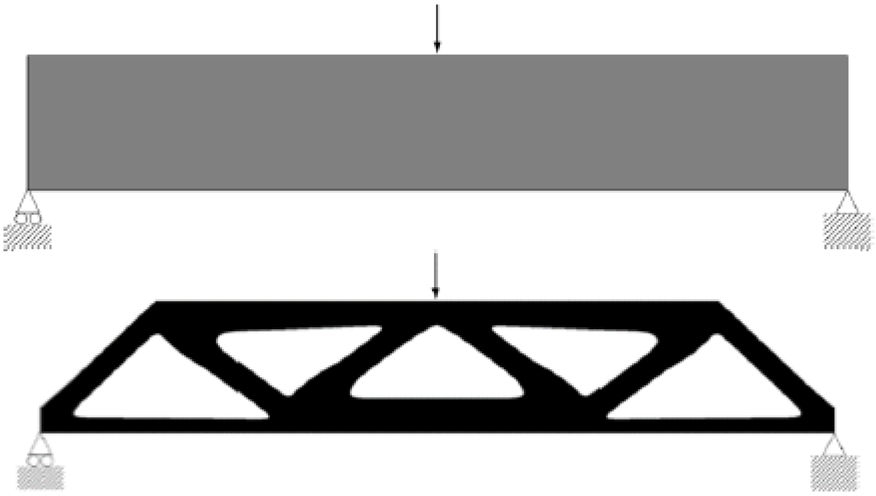


Figure 4: Topology optimization of MBB beam (Weihong et al., 2016)

2.2 Methods in Topology Optimization

Topology optimization encompasses a variety of methods that aim to optimize the material distribution within a given design domain. In general, topology optimization can be formulated as follows.

$$\begin{aligned} \underset{\rho}{\text{minimize}} \quad & f = f(\mathbf{u}(\rho), \rho) \\ \text{subject to} \quad & G_0(\rho) = \int_{\Omega} \rho \mathrm{d}V - V_0 \leq 0 \\ & G_j(\mathbf{u}(\rho), \rho) \leq 0 \text{ with } j = 1, \ldots, m \end{aligned}$$

Where,

$f(\mathbf{u}(\rho), \rho)$: The objective function to be minimized. Often in structural problems, the chosen objective is compliance. Reducing this compliance effectively enhances the structure's stiffness.

The distribution of material in the design domain is characterized by a binary density variable, with "0" denoting the absence and "1" signifying the presence of the material at any given point. This density variable, represented as ρ(x), designates whether the material is either existent (1) or non-existent (0) at each location within the domain.

Ω: The design space, signifies the permitted spatial region in which the design can materialize. $G_j(u(\rho), \rho) \leq 0$: The constraints that the solution is mandated to fulfill.

These could encapsulate various physical and design limitations, such as restricting the maximal material usage (volume constraints) or confining the stress values within the stipulated maxima.

2.2.1 Evolutionary Structural Optimization (ESO)

Evolutionary Structural Optimization (ESO) developed by Xie and Steven (Xie & Steven, 2012) is an approach that focuses on removing inefficient material from designs. It starts with a fully solid initial design and iteratively eliminates material based on its importance for structural performance. By gradually removing less critical material, ESO aims to find the most material-efficient configuration that maintains the desired structural performance.

The amount of material to be removed is steered by the Rejection Ratio (RRk), de_ned as

$$\frac{\sigma_e}{\sigma_{max}} < RR_k$$

where σ_e is the element stress and σ_{max} is the threshold stress.

2.2.2 Bi-directional Evolutionary Structural Optimization (BESO)

Bi-directional Evolutionary Structural Optimization (BESO), proposed by Querin and Young extends the concept of ESO by allowing for the addition of material as needed. BESO starts with an empty design domain and iteratively adds material to regions where it is structurally beneficial. This flexibility in material addition provides opportunities for innovative and efficient structural designs.

Mathematically, this is represented as:

$$\min f = \frac{1}{2} F^T u(x)$$

$$Subject\ to:$$

$$V^* - \sum_{i=1}^{n} V_i x_i = 0$$

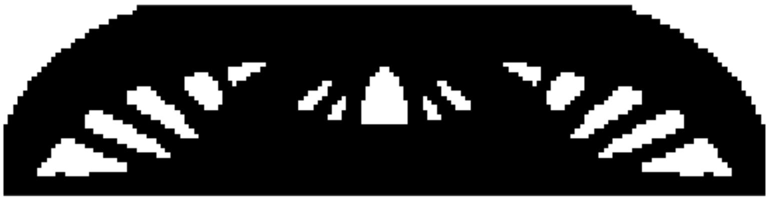
$$Where:$$

$x_i \in \{0,1\}$ for $i = 1, \dots, n$

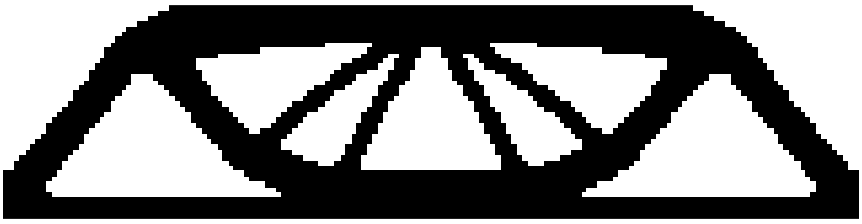
Where F is the vector of nodal forces, V* is the target volume and V_i is the volume of element i.

Examples of BESO after running beso.m file with different volume fractions :

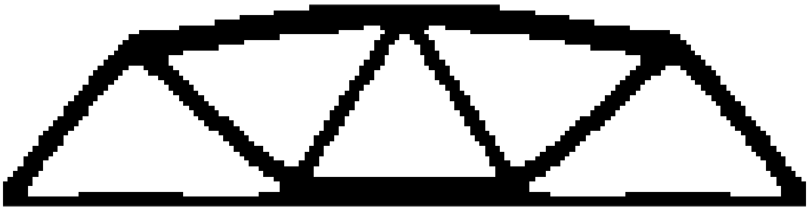
beso(80,40,0.8,0.02,2.5):



beso(80,40,0.5,0.02,2.5):



beso(80,40,0.3,0.02,2.5):



2.2.3 Density-Based Methods

Density-based methods play a significant role in topology optimization, including the widely used Solid Isotropic Material with Penalization (SIMP) method. These methods operate by penalizing the material in terms of element pseudo-density variables defined by a power-law.

$$E_i = E_0 \eta_i^p$$

Where, E_i represents the elastic modulus of the ith element, E_0 denotes the elastic modulus of the solid material, ηi corresponds to the pseudo density of the element, and p represents the penalty factor.(Weihong et al., 2016)

By iteratively adjusting these pseudo-density values, the optimization process seeks to find the optimal material distribution that maximizes structural performance while satisfying specified constraints. The penalization encourages the selection of either fully solid or void regions, resulting in material-efficient designs.

The problem is formulated as:

$$\text{minimize } f(x) = F^T u(x)$$

$$\mathit{Subjected\ to}:$$

$$V(x) = x^T V - V^* \leq 0$$

$$X = x \in \{x \in \mathbb{R}^n : 0 \leq x_i \leq 1\}$$

$$K(x)u(x) = F$$

Where V* is the target volume and V is the vector of volumes of all finite elements.

The results of running top99 code with different values of penalty is depicted below:

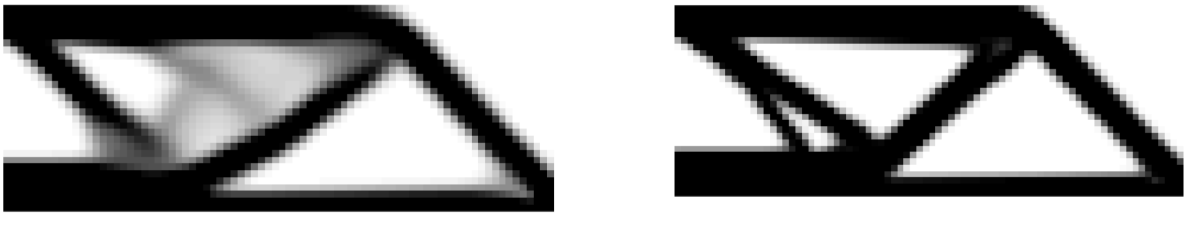


Figure 5: Classical half MBB topology optimization problem solved with SIMP and penal factor of 2 (left) and 5 (right). The solution id obtained by running top (80,30,0.5, penal, 2).

2.2.4 Level Set Methods

Level set methods provide a versatile framework for capturing and evolving geometric shapes during topology optimization. These methods represent the structural boundaries using a level set function, which describes the distance from each point in the domain to the boundary. By evolving the level set function, the material distribution is updated to optimize the objective function and satisfy the constraints. Level set methods allow for complex shape changes and topology variations during the optimization process, making them suitable for problems with evolving boundaries or multiple phases.

The level set function $\phi(x,y)$ describes the structure that occupies a domain Ω as:

$$\phi(x,y) = \begin{cases} \phi(x,y) = 0 \text{ if } (x,y) \in \partial\Omega \\ \phi(x,y) = 0 \text{ if } (x,y) \in \partial\Omega \\ \phi(x,y) > 0 \text{ if } (x,y) \notin \partial\Omega \end{cases}$$

Where: Ω signifies the domain occupied by the structure, ∂Ω denotes its boundary, and (x,y) corresponds to any point within this design domain.

The evolution equation for the level set function ϕ is :

$$\frac{\partial\phi}{\partial t} = v|\nabla\phi| - wg$$

Where: t represents time. $v(x,y)$ and $g(x,y)$ are scalar fields over the design domain

Ω. w is a positive parameter which determines the influence of the term involving g. ∇ϕ denotes the gradient of the level set function ϕ.

2.3 Machine Learning

Machine Learning is a rapidly advancing field that focuses on developing algorithms capable of learning patterns and making predictions based on data. This chapter provides a concise literature review of machine learning, covering fundamental concepts and key applications.

The chapter begins with an introduction to machine learning, including supervised, unsupervised, and reinforcement learning. It then explores Convolutional Neural

Networks (CNNs), a powerful deep learning architecture widely used for image and spatial data analysis. The integration of machine learning with topology optimization and the use of dimensionality reduction techniques, such as Principal Component Analysis (PCA), are also discussed.

By summarizing relevant studies, this chapter aims to provide a comprehensive understanding of the theoretical foundations and practical applications of machine learning. This knowledge will serve as a foundation for the subsequent chapters, where machine learning techniques will be implemented and evaluated in the context of topology optimization.

2.3.1 Introduction to Machine Learning and Deep Learning

Machine Learning (ML) and Deep Learning are two prominent branches of Artificial Intelligence (AI) that have revolutionized various industries and applications. ML focuses on developing algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed. As famously defined by Arthur Samuel, “Machine learning is the field of study that gives computers the ability to learn without explicitly being programmed”.

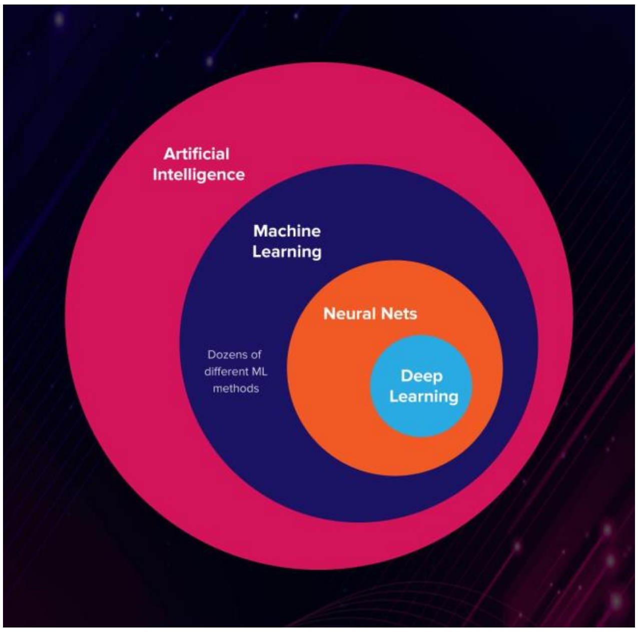


Figure 6: AI vs ML vs DL (Gavrilova, 2020)

In the broader landscape of AI, machine learning plays a crucial role as a subset. It encompasses a wide range of techniques, including supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves training models on labeled data, where the model learns to map inputs to corresponding outputs. Unsupervised learning, on the other hand, explores unlabeled data to discover patterns, structures, or relationships within the data. Reinforcement learning focuses on training agents to make sequential decisions based on feedback from the environment.

Deep Learning, a subset of machine learning, has gained significant attention and achieved remarkable success in recent years. Deep learning models are built using neural networks, computational models inspired by the structure and function of the human brain. These neural networks are composed of multiple layers, allowing them to learn complex representations of data. Deep learning has excelled in areas such as computer vision, natural language processing, and speech recognition, surpassing traditional machine learning approaches in many tasks.

2.3.4.1 Supervised Learning

Supervised learning is a fundamental category of machine learning where the algorithm learns from labeled training data. In supervised learning, the dataset consists of input samples along with their corresponding desired outputs or labels. The goal is to train a model that can accurately map inputs to outputs based on the provided labeled examples.

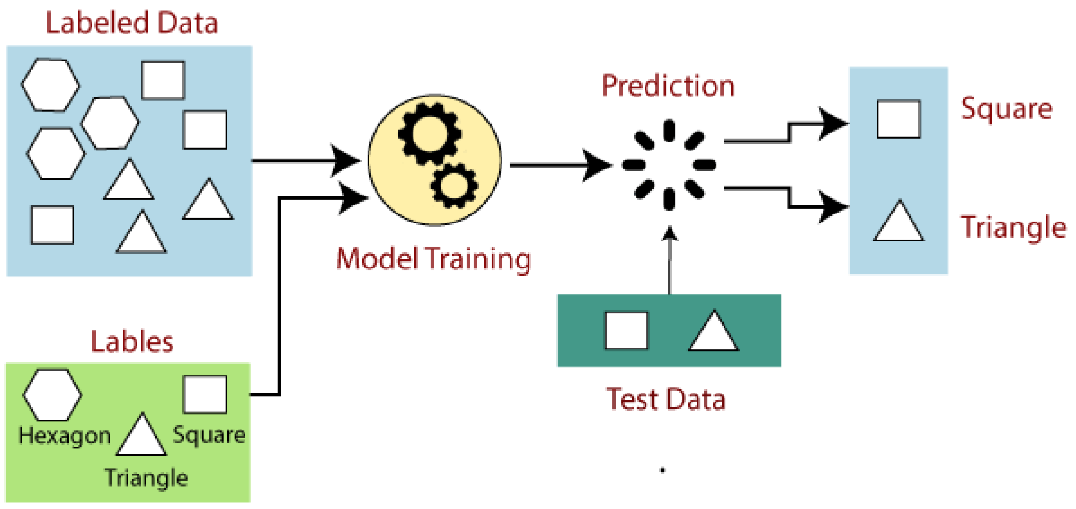


Figure 7: Supervised Learning (JavaTpoint, 2021)

The process of supervised learning involves training the model on the labeled data, where it learns to recognize patterns and relationships between the input features and the target outputs. This training phase involves adjusting the model's internal parameters to minimize the discrepancy between the predicted outputs and the true labels.

Once the model is trained, it can be used to make predictions on new, unseen data. By generalizing from the training examples, the model can infer the correct output for input samples that it has not encountered before.

2.3.4.2 Unsupervised Learning

Unsupervised learning is a branch of machine learning where the algorithm learns from unlabeled data, without any explicit target or output labels. Unlike supervised learning, where the goal is to map inputs to outputs, unsupervised learning focuses on discovering patterns, structures, and relationships within the data itself.

In unsupervised learning, the algorithm explores the inherent structure of the data and identifies meaningful patterns or clusters. It does so by extracting relevant features and organizing the data based on similarities or dissimilarities between samples. This process is often referred to as clustering or dimensionality reduction.

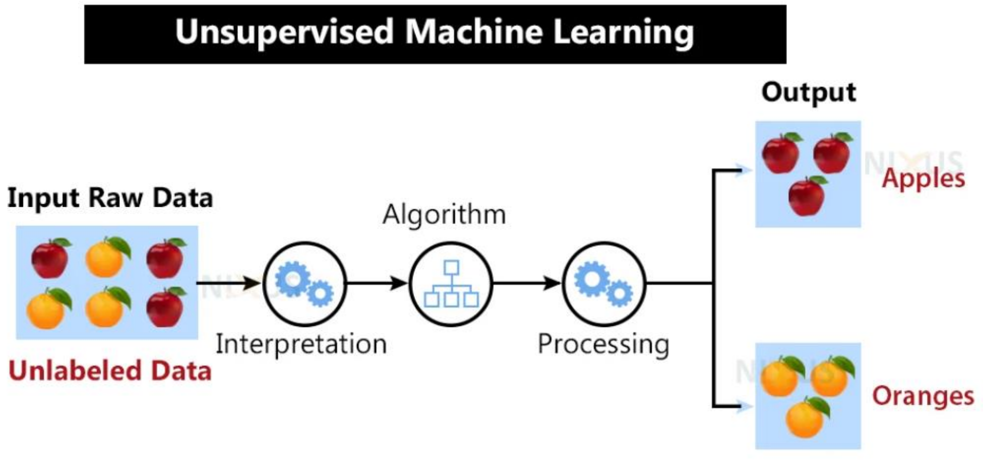


Figure 8: Unsupervised Learning (NIXUS, n.d.)

2.3.4.3 Reinforcement Learning

Reinforcement learning is a type of machine learning where an agent learns to make decisions and take actions in an environment to maximize cumulative rewards. Unlike supervised and unsupervised learning, reinforcement learning operates in a dynamic setting where the agent interacts with the environment, receives feedback in the form of rewards or penalties, and learns through a trial-and-error process.

In reinforcement learning, the agent learns by exploring the environment, taking actions, and observing the resulting state transitions and associated rewards. The goal is to find an optimal policy that guides the agent's decision-making process to maximize the long-term cumulative rewards.

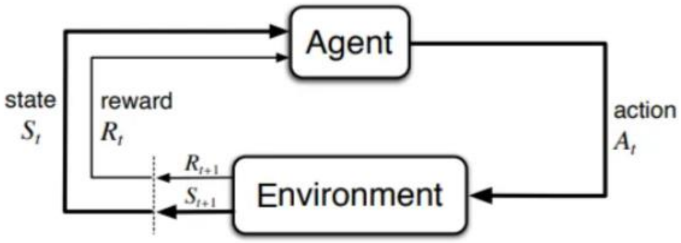


Figure 9: Reinforcement Learning Cycle (blackburn, 2019)

2.3.2 Deep Learning

Deep learning is a subfield of machine learning that focuses on training artificial neural networks with multiple layers to learn and represent complex patterns and relationships in data. Unlike traditional machine learning algorithms that rely on handcrafted features, deep learning algorithms have the ability to automatically learn hierarchical representations of data from raw input.

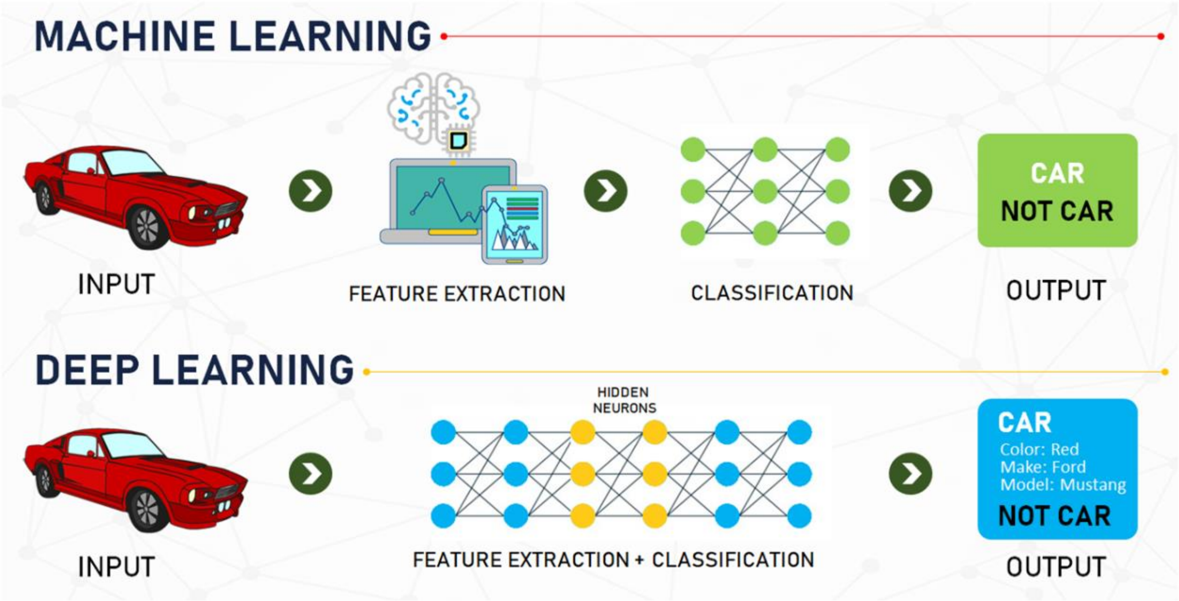


Figure 10: Deep learning in layman's terms (Labs, 2020)

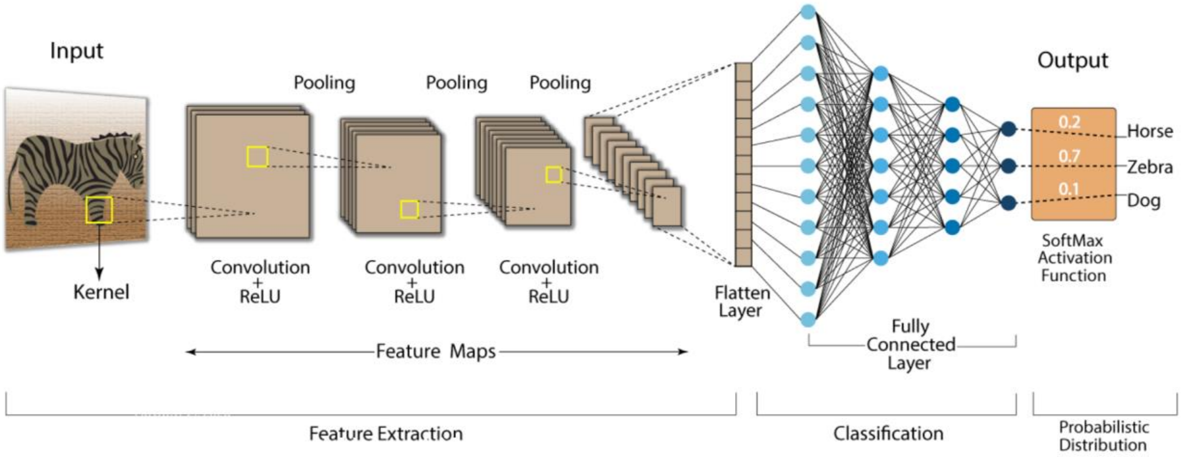
At the heart of deep learning are neural networks, which are computational models inspired by the structure and function of biological neurons in the human brain. Neural networks consist of interconnected nodes called neurons, organized in layers. The input layer receives the raw data, and subsequent hidden layers extract progressively more abstract and high-level features. The final output layer produces the desired predictions or classifications. (Saxena, 2017)

2.3.4.4 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a powerful class of deep neural networks that have achieved remarkable success in image recognition, computer vision, and other spatial data analysis tasks. CNNs are specifically designed to handle grid-like data, such as images, by leveraging their unique architectural properties.

The key idea behind CNNs is the use of convolutional layers, which apply filters or kernels to local regions of the input data. These filters capture spatial patterns and features, allowing the network to learn hierarchical representations of increasing complexity. The convolution operation involves element-wise multiplication of the filter with the input followed by summing the results, which produces a feature map.

CNNs typically consist of multiple convolutional layers, interspersed with other layers like pooling layers, activation functions, and fully connected layers. Pooling layers downsample the feature maps, reducing their spatial dimensionality while preserving important features. Activation functions introduce non-linearities into the network, enabling the modeling of complex relationships. Fully connected layers connect every neuron from one layer to the next, allowing for high-level abstraction and decision-making. (IBM, n.d.)



One of the main advantages of CNNs is their ability to automatically learn hierarchical representations from raw input data, eliminating the need for manual feature engineering. This makes CNNs well-suited for image recognition tasks, where detecting local patterns and capturing spatial dependencies are crucial. By learning hierarchical representations, CNNs can effectively recognize objects, segment images, and perform various other image analysis tasks.

In the context of topology optimization, CNNs have been successfully applied to analyze and classify structural data. They can learn to identify structural elements, detect anomalies, and predict the behavior of different designs based on their visual representations. CNNs excel at extracting spatial features and patterns, enabling them to analyze structural data efficiently. (Yiquan Zhang, 2020)

2.3.4.5 Dimensionality Reduction Techniques

Dimensionality reduction techniques play a crucial role in machine learning and data analysis, including applications in structural optimization. These techniques aim to reduce the dimensionality of high-dimensional data while preserving its important characteristics. By reducing the number of variables or features, dimensionality reduction methods simplify the analysis and interpretation of the data, improve computational efficiency, and mitigate the risk of overfitting.

One widely used dimensionality reduction technique is Principal Component Analysis (PCA). PCA transforms the original high-dimensional data into a new set of orthogonal variables called principal components. These components capture the maximum variance in the data, allowing for a more concise representation of the information. By selecting a subset of the principal components, the dimensionality of the data can be significantly reduced while preserving a large portion of its variability. (Jolliffe, 2006)

3 Methodology and Implementation

3.1 Dataset

Datasets are pivotal in deep learning, providing the foundation for training and validating neural networks. For topology optimization, the TOP dataset is invaluable. It contains solutions for 10,000 randomly generated scenarios, each detailing 100 iterations on a 40×40 grid. Created using the Topy framework and the SIMP method, it offers a comprehensive view of the optimization process, ensuring robust neural network training and deeper insights into topology optimization challenges.

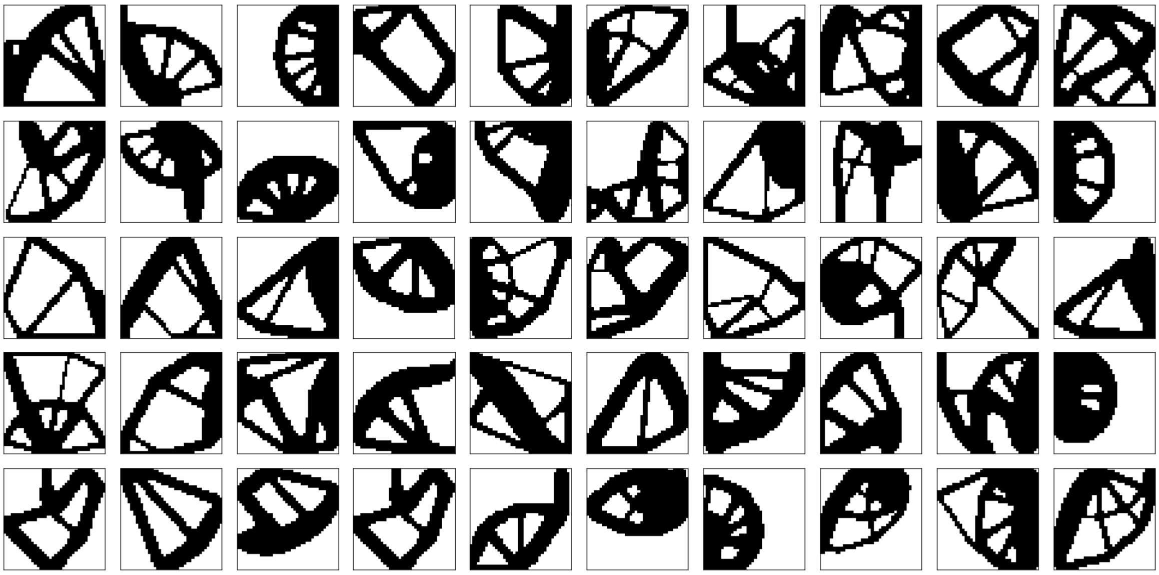


Figure 11: The samples from the TOP dataset. (Sosnovik, 2017)

Dataset Characteristics:

- Shape and Dimensions: Each entry in the dataset is a tensor with the shape (100, 40, 40). This represents 100 iterations of the optimization process on a 40×40 grid. The grid captures the spatial distribution of the material in each iteration, providing a step-by-step evolution of the topology optimization process.

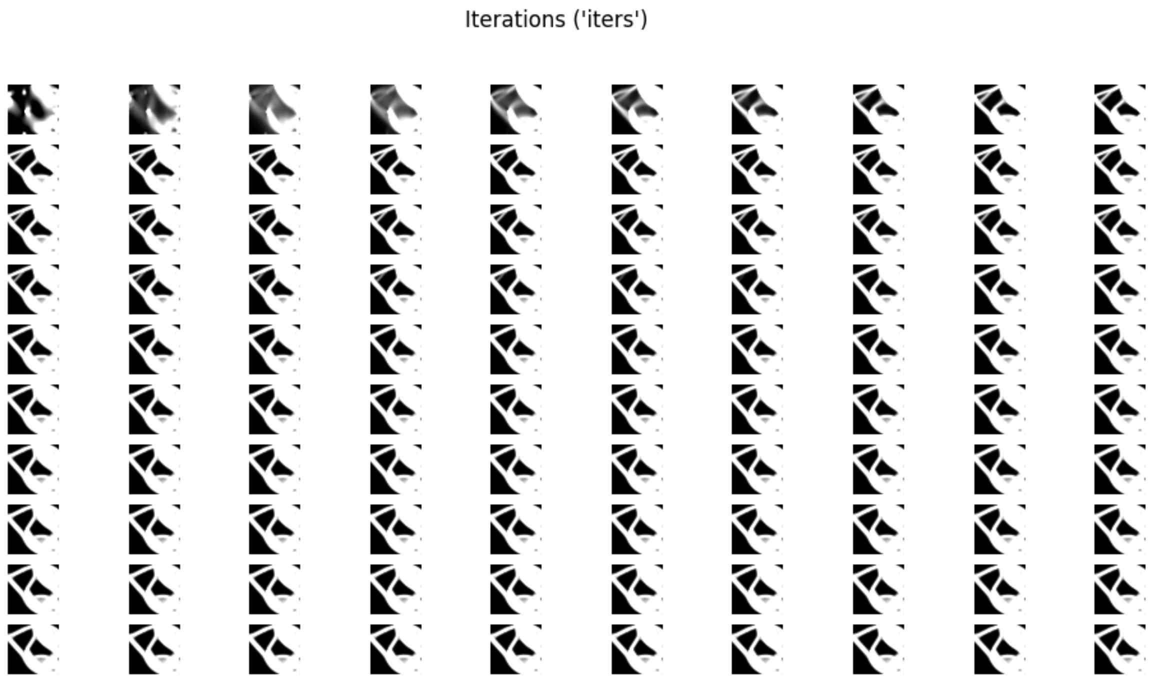


Figure 12: Example of 100 iterations of a sample from 10000 samples

- Generation Methodology: The dataset was meticulously generated using the Topy framework, which employs the SIMP (Solid Isotropic Material with Penalization) approach to topology optimization.
- Constraints and Loads: The constraints and loads for each problem in the dataset were determined through a systematic process:
  - Nodes with fixed x and y translations, as well as the number of loads, are sampled from a Poisson distribution.
  - The nodes for these constraints are chosen based on a distribution defined on the grid. Notably, boundary nodes have a 100 times higher probability of being selected compared to inner nodes.
  - All load values are set to -1.
  - The volume fraction, which represents the ratio of the material's volume to the total volume, is sampled from a Gaussian distribution. This distribution has a mean of 0.5 and a standard deviation of 0.1, ensuring a balanced representation of different volume fractions.

3.2 Neural Network Architectures

For the evaluation, multiple neural network architectures were explored. These architectures vary in terms of the number of encoder and decoder layers, channel counts, and optimizer types.

a) Encoder-Decoder Net Model:

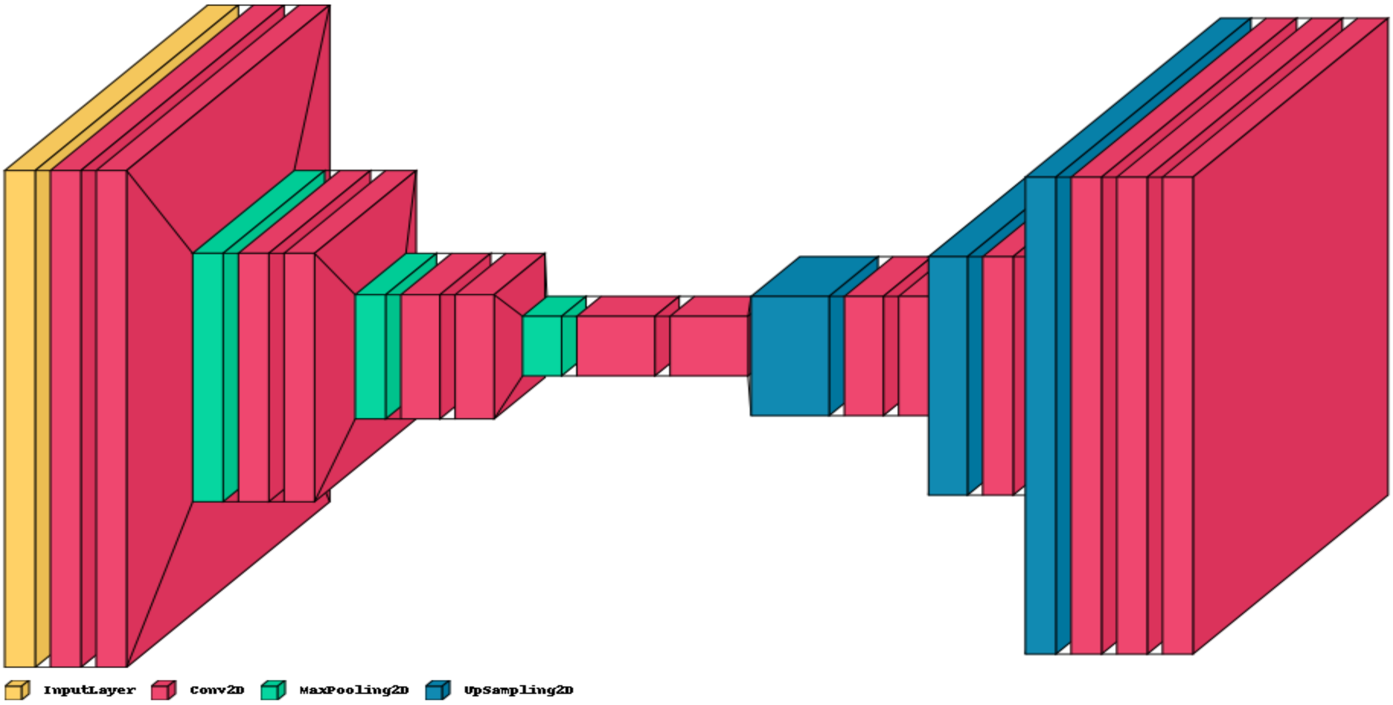


Figure 13: Encoder-Decoder Net Model

The Encoder-Decoder Net model is a sophisticated deep convolutional neural network, drawing inspiration from the U-Net architecture. This model is adeptly designed for tasks that necessitate the extraction of both localized and overarching information from input images, ensuring a comprehensive approach to image processing.

The architecture commences with the encoder, which is instrumental in capturing the hierarchical features of the input image. This is achieved through a series of blocks. The initial block, termed 'encoder1', incorporates two convolutional layers, each with a 3x3 kernel size, transitioning from 100 input channels to 64 output channels. These layers are succeeded by ReLU activations, and the output is subsequently downsampled using a MaxPooling layer with a 2x2 kernel. The subsequent block, 'encoder2', employs two convolutional layers transitioning from 64 input channels to

128 output channels, each followed by ReLU activations. The output from this block undergoes further downsampling via a MaxPooling layer. The third encoder block, 'encoder3', mirrors its predecessors but operates on 128 input channels, producing 256 output channels. Post convolution, the output is subjected to another round of downsampling through a MaxPooling layer.

Centrally positioned is the middle block, which refines the features extracted by the encoder. It houses two convolutional layers, processing 256 input channels and delivering 512 output channels, with each convolution succeeded by a ReLU activation.

The decoder pathway, symmetric to the encoder but inverse in function, focuses on upscaling and reconstructing the output. The initial decoder block upscales the feature map from 512 channels to 256 and then processes it through two convolutional layers, yielding 256 output channels. The subsequent block further upscales the feature map to 128 channels and refines it through two convolutional layers, producing 128 output channels. The final block in this trajectory upscales the feature map to 64 channels, which is then processed through two convolutional layers, resulting in 64 output channels.

Concluding the architecture is the output layer. This pivotal layer consolidates the rich 64 channels from the decoder into a singular output channel using a convolutional operation with a 1x1 kernel, ensuring a concise and representative output.

b) Smaller Network:

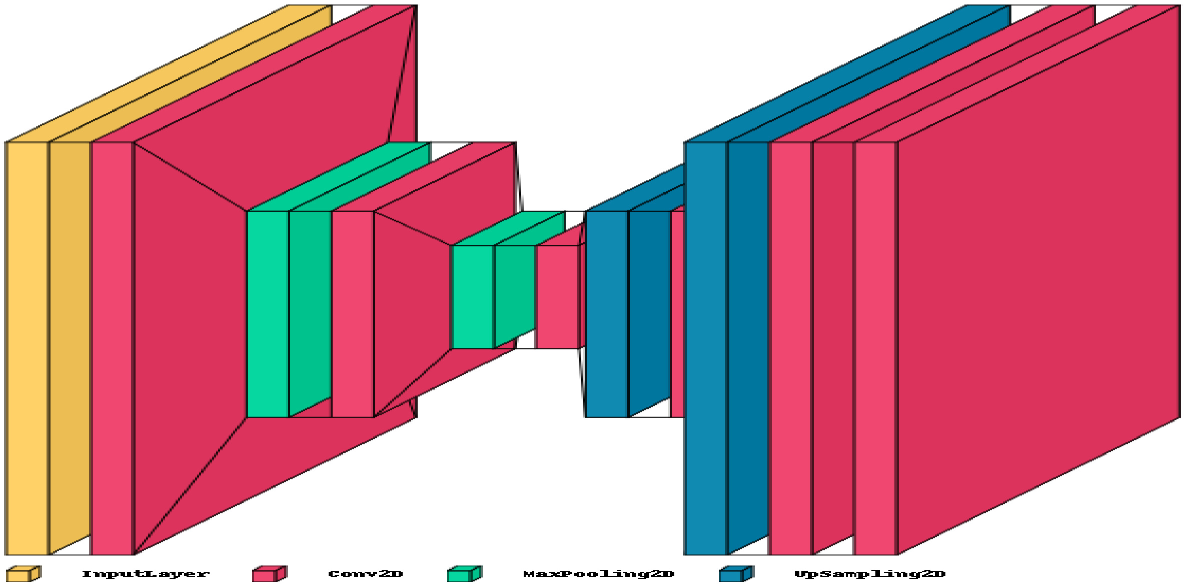


Figure 14: Smaller Network

The Smaller Network is a streamlined convolutional neural network, taking cues from the U-Net architecture. While it retains the foundational principles of the U-Net, it distinguishes itself with a reduced number of layers, offering a more compact design.

Initiating with the encoder, the network captures hierarchical features from the input image. The encoder's first block comprises two convolutional layers with a 3x3 kernel size, transitioning from 100 input channels to 32 output channels. These layers are complemented by ReLU activations, and the resultant output undergoes downsampling via a MaxPooling layer. The subsequent block in the encoder houses two convolutional layers, processing 32 input channels to yield 64 output channels. Each of these layers is succeeded by ReLU activations, and the output is further refined through another MaxPooling layer.

Positioned centrally is the middle block, which serves to refine the encoder's features. It employs two convolutional layers, processing 64 input channels and delivering 128 output channels, with each layer followed by a ReLU activation.

The decoder pathway, designed for upscaling and reconstruction, is composed of two primary blocks. The initial block upscales the feature map from 128 channels to 64 and refines it through two convolutional layers, resulting in 64 output channels. The subsequent block further upscales the feature map to 32 channels and processes it through two convolutional layers, yielding 32 output channels.

Concluding the architecture, the output layer consolidates the 32 channels from the decoder into a singular, representative output channel using a convolutional operation with a 1x1 kernel.

c) Tiny Neural Network:

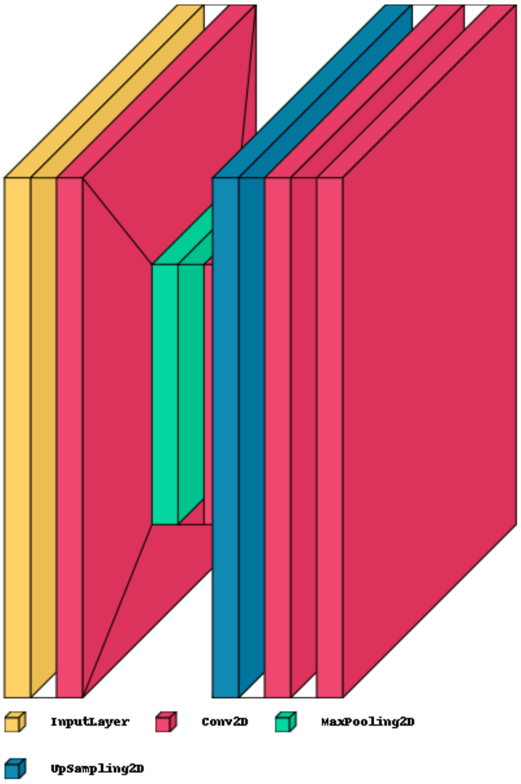


Figure 15: Tiny Neural Network

The Tiny Neural Network, inspired by the U-Net architecture, is tailored for lightweight tasks. Its encoder features a block with two convolutional layers, transitioning from 100 to 16 channels, each with a 3x3 kernel and ReLU activations, followed by a MaxPooling layer. The central section refines these features through two convolutional layers, processing 16 input channels to yield 32 output channels, each accompanied by a ReLU activation. The decoder, in turn, upscales the feature map from 32 channels back to 16, refining it through two convolutional layers. The architecture culminates with an output layer that merges the 16 channels into a singular output channel using a 1x1 convolutional kernel.

3.3 Training of Neural Networks

3.3.1 Dataset Partitioning:

The Topology Optimization Dataset (TOP) was systematically partitioned into three distinct subsets: training, validation, and testing. The rationale behind this partitioning is rooted in the principles of machine learning, where distinct data subsets are essential for different phases of model development:

Training Subset (70% of the data): This subset is primarily used to adjust the model's weights and biases. The majority allocation of the dataset to this subset ensures that the neural network has a comprehensive set of examples to learn the underlying patterns and relationships.

Validation Subset (20% of the data): Post the initial training phase, the model is fine-tuned using the validation subset. This subset aids in hyperparameter optimization and helps in mitigating overfitting by providing an unbiased evaluation of the model's performance during the training phase.

Testing Subset (10% of the data): Reserved for the final phase, the testing subset provides an objective assessment of the model's predictive capabilities. It offers insights into how the trained model will perform on entirely unseen data, ensuring that the evaluation is free from any biases introduced during the training or validation phases.

3.3.2 Optimization Strategy:

For the optimization of the neural network's weights and biases, the Adam optimizer was employed. Adam optimizer, short for Adaptive Moment Estimation, is an optimization algorithm that computes adaptive learning rates for each parameter. It combines the advantages of two other extensions of stochastic gradient descent: AdaGrad and RMSProp.

To investigate the sensitivity of the model's performance to the learning rate, multiple experiments were conducted with varying learning rates. This systematic approach provides a comparative analysis, shedding light on the optimal learning rate that yields the highest prediction accuracy.

4 Results

4.1 Prediction Performance Across Different Architectures

To ascertain the efficacy of various neural network architectures in the context of the Topology Optimization Dataset (TOP), a comprehensive evaluation was undertaken. Each architecture was trained using the same dataset partitioning strategy, ensuring consistency in the evaluation process.

The prediction results, juxtaposed with the corresponding loss vs. epoch graphs, provide a visual representation of the learning trajectory of each architecture. These graphs elucidate the convergence behavior, indicating the epochs where the loss plateaus, which is indicative of the model reaching its optimal performance.

a) Encoder Decoder Net Model

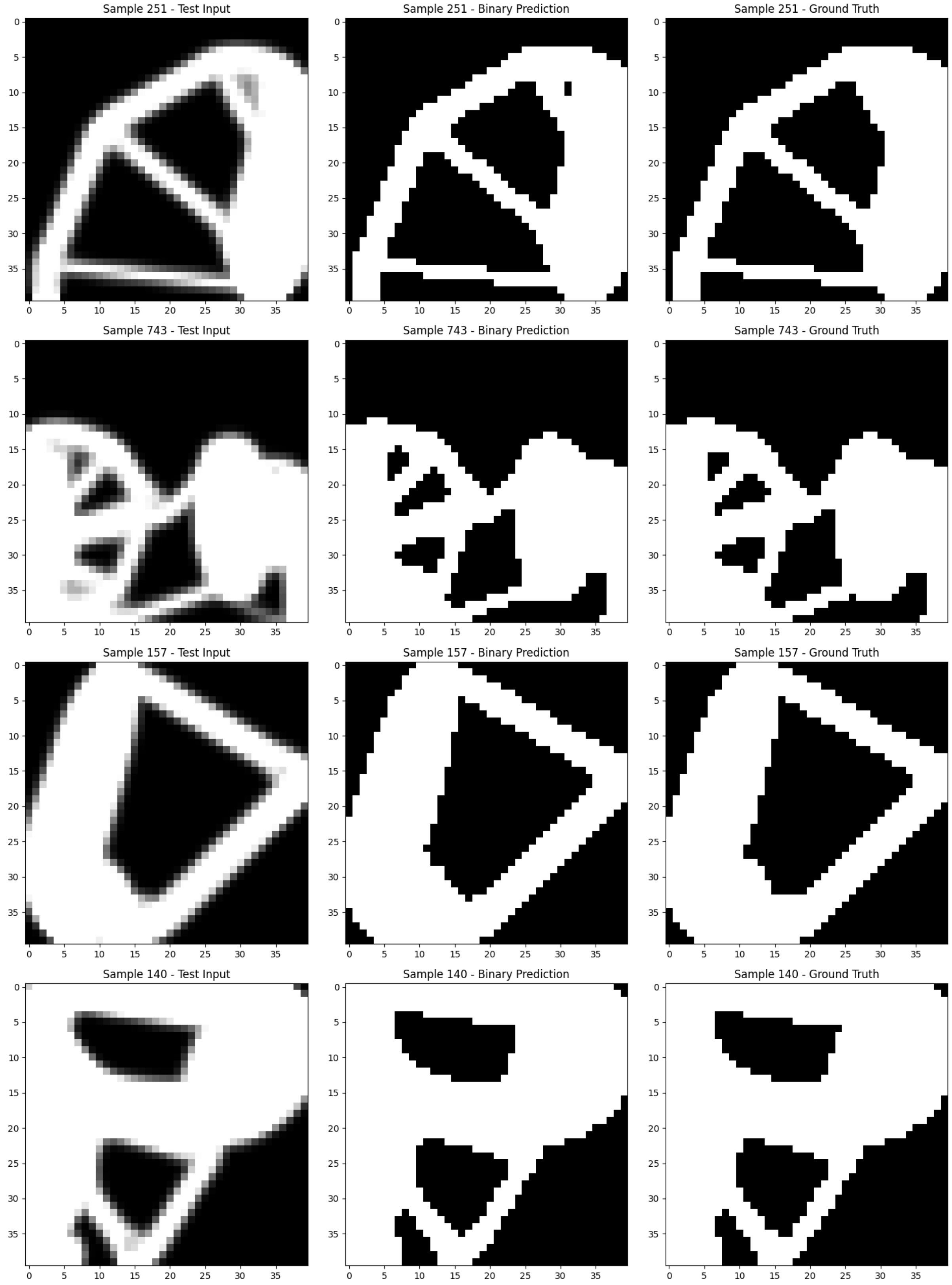


Figure 16: Comparison of Input, Prediction and Ground Truth- Encoder Decoder Net Model

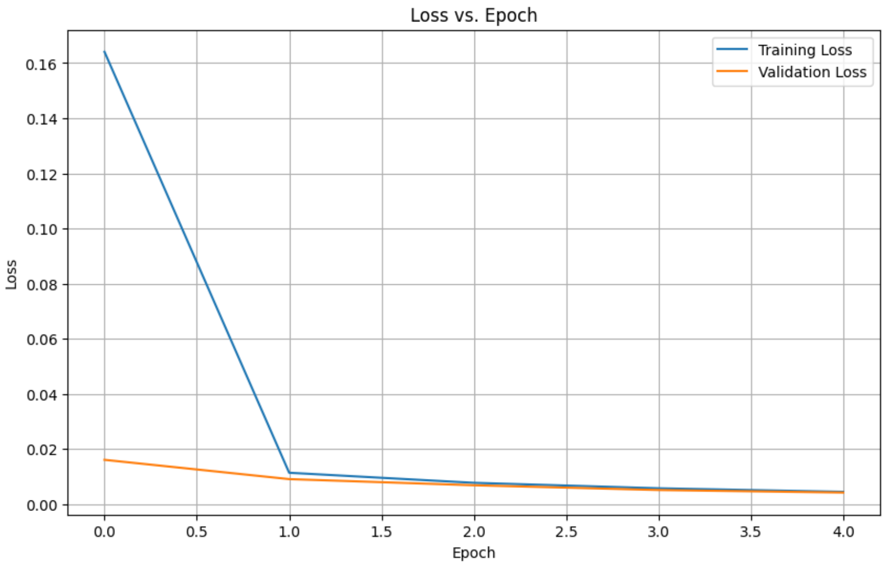


Figure 17: Encoder Decoder Model- Loss vs. Epochs Plot

b) Smaller Network

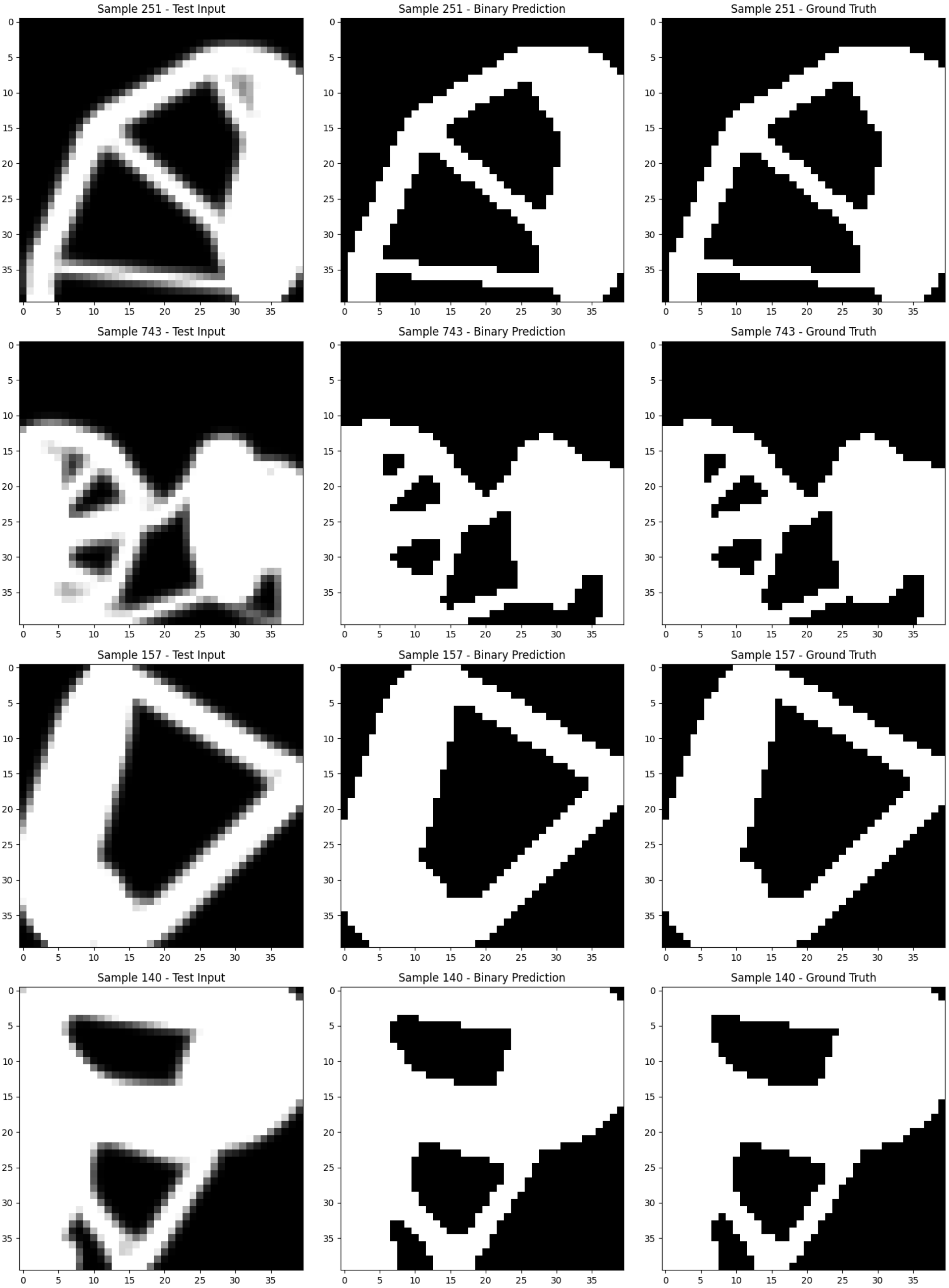


Figure 18: Comparison of Input, Prediction and Ground Truth- Smaller Model

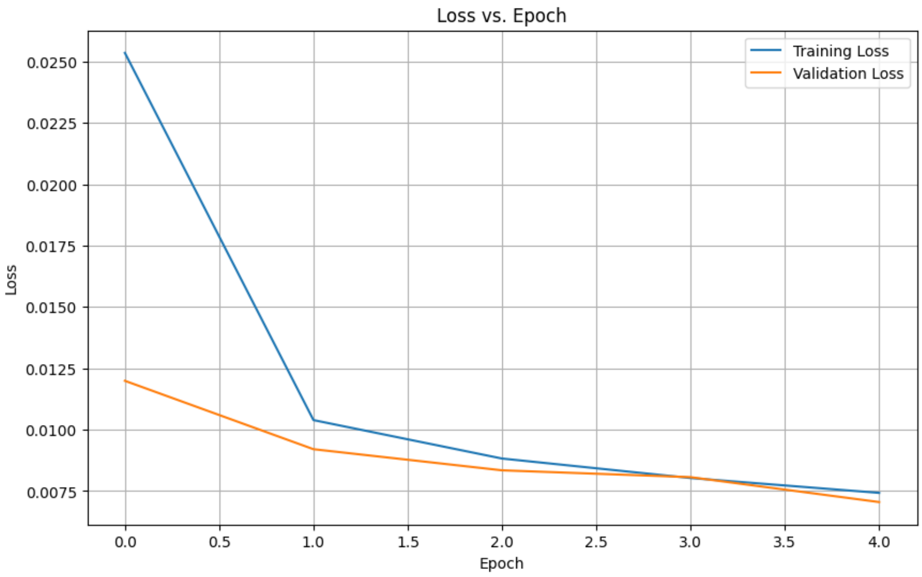


Figure 19: Smaller Model- Loss vs. Epochs Plot

c) Tiny Network

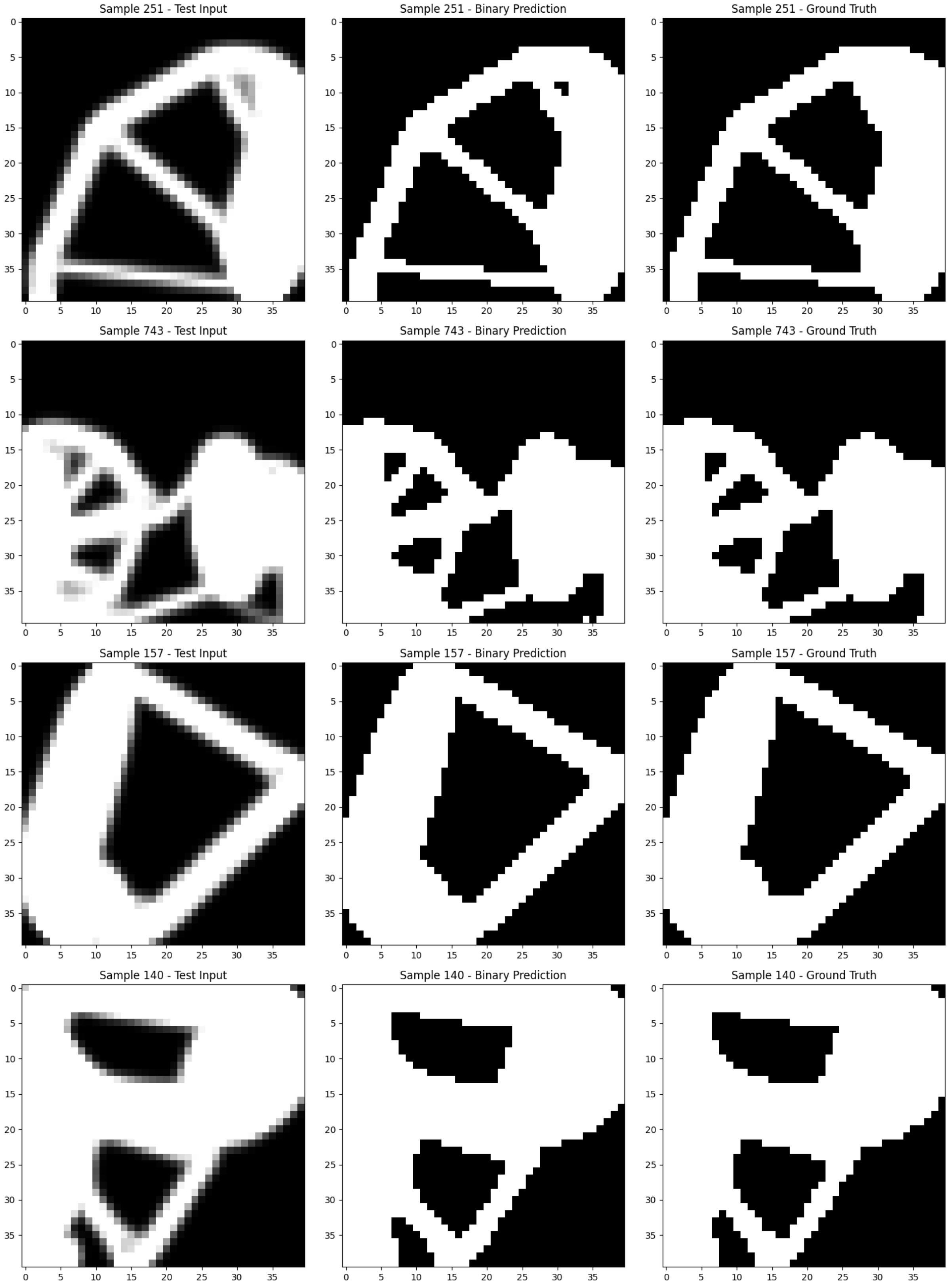


Figure 20: Comparison of Input, Prediction and Ground Truth- Tiny Network Model

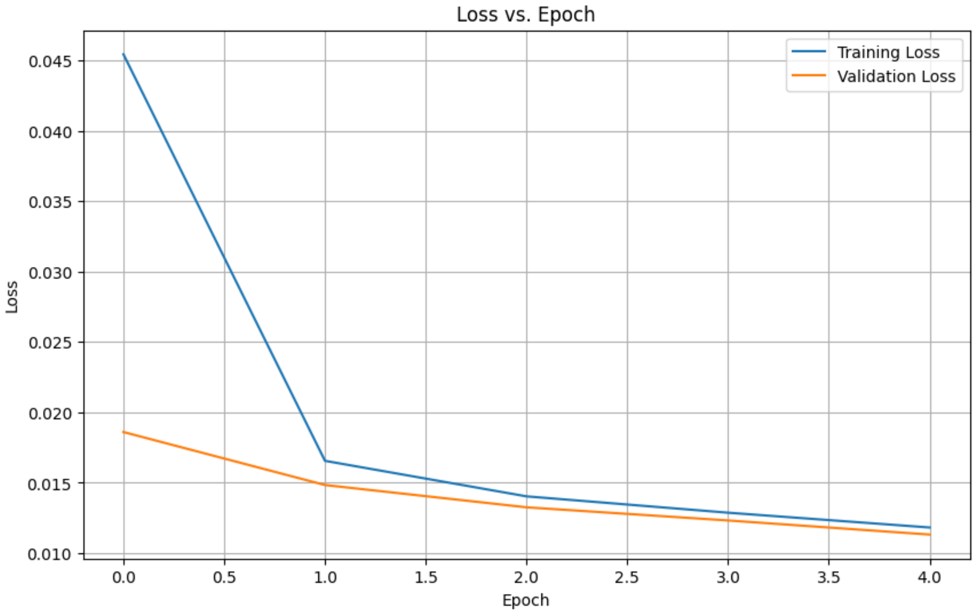


Figure 21: Tiny Network Model- Loss vs. Epochs Plot

The comprehensive evaluation of various neural network architectures on the Topology Optimization Dataset (TOP) has yielded insightful findings. A salient observation is the remarkable accuracy of the model's predictions, which closely align with the ground truth. This underscores the model's capability to capture intricate patterns and relationships within the dataset, thereby facilitating precise predictions.

However, a nuanced analysis reveals that as the complexity of the neural network architecture diminishes, there is a slight deviation in the predictions. Simpler or less intricate models, while still effective, exhibit a marginally reduced fidelity in their predictions compared to their more complex counterparts. This suggests a trade-off between model complexity and prediction accuracy, emphasizing the importance of selecting an architecture that strikes a balance between computational efficiency and predictive precision.

Furthermore, the loss vs. epoch graphs for all evaluated architectures consistently demonstrate that with an increase in the number of epochs, both training and validation losses experience a decline. This trend is emblematic of the models' learning progression, where they continually refine their weights and biases to minimize the discrepancy between predicted and actual values. The observed plateauing of the loss at certain epochs indicates the models' convergence to their optimal performance levels.

4.2 Analysis of Loss Across Varying Learning Rates for the Encoder-Decoder Architecture

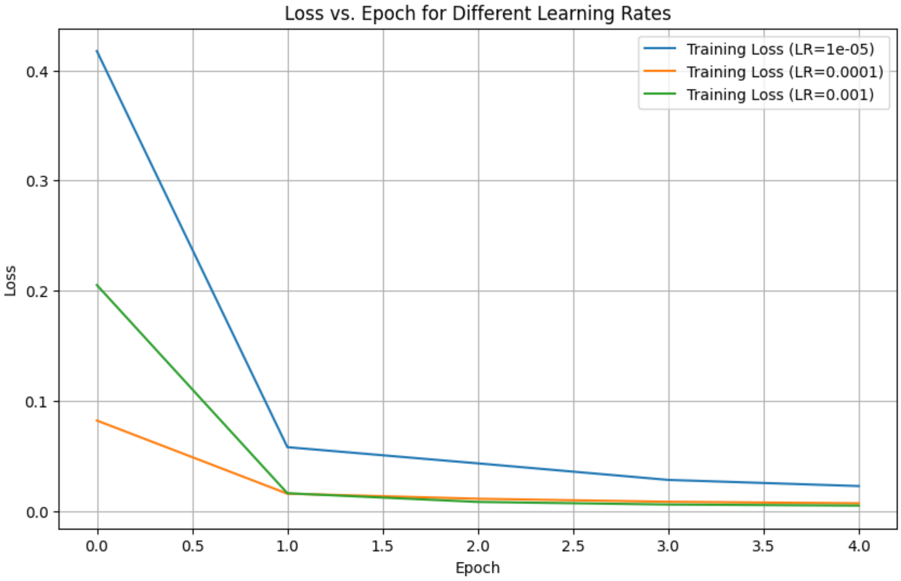


Figure 22: Training Loss vs. Epoch- Encoder Decoder Model

The analysis of loss across varying learning rates for the Encoder-Decoder architecture provides critical insights into the model's convergence behavior and the impact of the learning rate, a pivotal hyperparameter, on the training process.

From the presented plots, it is evident that all three learning rates—1e-5, 0.0001, and 0.001—lead to a decrease in loss as the number of epochs increases, signifying convergence. This consistent decline across different learning rates underscores the model's ability to learn and adapt, regardless of the initial learning rate setting. However, the rate of convergence and the final loss value attained vary significantly based on the chosen learning rate.

Specifically, a learning rate of 1e-5 results in the highest loss among the three, indicating a slower convergence. This can be attributed to the smaller step sizes taken during the optimization process, which, while ensuring a more meticulous search, might require more epochs to reach the optimal solution. On the other hand, a learning rate of 0.0001 yields the lowest loss, suggesting a more rapid and efficient convergence to a potentially better local minimum. The learning rate of 0.001, while faster than 1e-5, does not achieve as low a loss as 0.0001, positioning it in an intermediate state between the two extremes.

In essence, the selection of an appropriate learning rate is paramount to the training efficacy and model performance. While a lower learning rate ensures a detailed exploration of the loss landscape, it might be computationally expensive and slow. Conversely, a moderately higher learning rate can strike a balance between convergence speed and accuracy, as evidenced by the superior performance of the 0.0001 learning rate in this analysis. This underscores the importance of meticulous hyperparameter tuning to achieve optimal model performance in the realm of neural network training.

5 Conclusion

The integration of neural network architectures with the Topology Optimization Dataset (TOP) illuminates a pathway towards enhancing understanding in the realm of topology optimization through advanced computational methodologies. This research underscores the pivotal role of strategic model design and meticulous hyperparameter tuning in optimizing predictions and highlights the potential of modern computational techniques in advancing topology optimization.

Findings from the systematic exploration of various architectures and a spectrum of learning rates provide a comprehensive perspective on model convergence, accuracy, and performance, demonstrating that the efficacy of neural networks is significantly influenced by decisions made during the model-building phase.

Outlined future work, which includes the analysis of diverse datasets, application of transfer learning, and exploration of different input resolutions, underscores the expansive and yet-to-be-explored horizons in the integration of machine learning and deep learning techniques in topology optimization. These avenues not only signify the depth and breadth of possibilities within the realm of machine learning and deep learning but also emphasize their transformative potential in redefining the paradigms of topology optimization.

In conclusion, the intersection of traditional engineering design and innovative computational techniques accentuates the transformative potential of this confluence. The amalgamation of established optimization methodologies with the dynamic capabilities of neural networks is poised to catalyze a paradigm shift, heralding a future where design solutions are not only efficient but also intrinsically innovative.

6 Further Work

The results derived from the current study offer valuable insights into the performance of neural network architectures on the Topology Optimization Dataset (TOP). To further enhance the models' robustness and applicability, several areas of exploration and refinement are proposed:

6.1 Exploration with Varied Datasets

Expanding the dataset to encompass a broader range of load cases, constraints, and geometric shapes can provide a more holistic understanding of the model's capabilities. By simulating a diverse array of real-world scenarios, the model will be challenged to generalize across different topological configurations. Evaluating the model's predictive accuracy, loss convergence, and computational efficiency across these varied datasets will be crucial. This comprehensive evaluation will ascertain the model's adaptability and versatility in handling diverse topological scenarios.

6.2 Enhancement through Transfer Learning

Transfer learning offers the potential to significantly enhance the quality and efficiency of topology optimization predictions. By leveraging models that have been pre-trained on related tasks or on larger datasets, we can provide a foundational knowledge base for the current task. This approach can potentially lead to faster convergence and improved prediction outcomes. Once these pre-trained models are integrated, they can be fine-tuned to the specific characteristics of the TOP dataset. This ensures that the model is not only benefiting from generalized knowledge but is also tailored to the unique intricacies of topology optimization.

6.3 Investigating Different Aspect Ratios of Input Data

The resolution of input data can have a profound impact on a model's predictive capabilities and computational demands. By experimenting with input data of varying aspect ratios, beyond the current 40x40 image size, we can delve deeper into the model's adaptability. Higher resolutions might enable the model to discern finer topological details, while lower resolutions present the challenge of extracting essential features from more generalized data. Assessing the model's performance across these different resolutions will shed light on the optimal balance between data detail and computational efficiency. This exploration will provide a clearer understanding of the trade-offs involved and guide data preprocessing decisions in future studies.

By pursuing these areas of further research, we aim to refine our understanding and push the envelope in the realm of neural network-based topology optimization.

References

1. Bendsϕe, M. P. (2004). *Topology Optimization: Theory, Methods and Applications.* Springer.

2. blackburn. (2019, July 18). *Reinforcement Learning : Markov-Decision Process (Part 1)*. Retrieved from Towards Data Science: https://towardsdatascience.com/introduction-to-reinforcement-learning-markov-decision-process-44c533ebf8da

3. Gavrilova, Y. (2020, April 8). *Artificial Intelligence vs. Machine Learning vs. Deep Learning: Essentials*. Retrieved from https://serokell.io/blog/ai-ml-dl-difference

4. IBM. (n.d.). *Convolutional Neural Networks*. Retrieved from https://www.ibm.com/topics/convolutional-neural-networks

5. *JavaTpoint*. (2021). Retrieved from https://www.javatpoint.com/supervised-machine-learning

6. Labs, D. (2020, March 2). *Understanding Machine Learning & Deep Learning*. Retrieved from DLT Labs: https://dltlabs.medium.com/understanding-machine-learning-deep-learning-f5aa95264d61

7. NIXUS. (n.d.). *Unsupervised Learning Types, Algorithms and Applications*. Retrieved from https://nixustechnologies.com/unsupervised-machine-learning/

8. Peter W. Christensen, A. K. (2009). *An Introduction to Structural Optimization.* Springer.

9. Saxena, S. (2017, October 26). *Artificial Neuron Networks(Basics)* | *Introduction to Neural Networks*. Retrieved from Becoming Human: Artificial Intelligence Magazine: https://becominghuman.ai/artificial-neuron-networks-basics-introduction-to-neural-networks-3082f1dcca8c

10. Sosnovik, I. a. (2017). *Top*. Retrieved from Github: https://github.com/ISosnovik/top

11. Yiquan Zhang, B. P. (2020). *A deep Convolutional Neural Network for topology optimization with strong generalization ability.* arXiv.